COL724 A3

Divyanka Chaudhari (2019CS50429)

Part 1

Controller Hub
It mimics the behavior of an Ethernet hub. When it receives a packet (a frame, at layer 2 of the OSI model), it doesn't try to figure out the best place to send it. Instead, it simply floods the packet out of all ports on the switch, except for the port that it arrived on. This means every machine on the network gets the packet, regardless of whether it's the intended recipient or not. It's not efficient because it creates a lot of unnecessary network traffic.

In the context of an SDN controller like Ryu, the Controller Hub is implemented as an application that listens for "packet-in" events (which indicate that the switch doesn't know where to send a packet). For each of these events, the Hub Controller responds by telling the switch to send the packet out through all its other ports.

Learning Switch
It's more like a traditional network switch. It tries to learn the layout of the network by remembering the source MAC address and the switch port of incoming packets. This way, when it receives a packet, it looks at the destination MAC address and determines if it knows which port leads to that address. If it does, it sends the packet out only on that port. If it doesn't, it floods the packet, just like a hub.

Over time, as it learns the locations of different MAC addresses, it becomes more efficient because it doesn't need to flood the network with unnecessary copies of each packet.

In the context of an SDN controller, the Learning Switch is implemented as an application that maintains a table mapping MAC addresses to ports. When a "packet-in" event occurs, the Learning Switch application checks its table to see if it knows which port to use for the destination MAC address. If it does, it tells the switch to send the packet to that port. If it doesn't, it floods the packet and adds the source MAC address and port to its table for future use.

Implementation in Ryu
Both the Controller Hub and Learning Switch are implemented as standalone Ryu applications. They are run using the ryu-manager command, which starts the Ryu controller and loads the specified application. The controller then communicates with the switch using the OpenFlow protocol, sending it instructions on how to handle network traffic.

These two contrasting approaches demonstrate the difference between an unmanaged network (the hub) and a managed network(the switch).

Part 2

Firewall Functionality
The firewall functionality is based on predefined rules that determine which traffic should be blocked. These rules are defined by the MAC addresses of the source and destination hosts. If a packet's source and destination MAC addresses match any of the specified rules, the packet is dropped, effectively blocking the communication between those hosts.

- **Rule Installation:** The firewall rules are not installed proactively in this script. Instead, the application checks each incoming packet against the firewall rules in the `packet_in_handler` method.
- **Packet Handling:** When a packet arrives at the switch and there's no flow entry for it, the switch sends a packet-in event to the controller. The controller's `packet_in_handler` method processes this event. If the source and destination MAC addresses match a firewall rule, the packet is dropped. This is done by simply not sending a packet-out message or installing a corresponding flow entry for the packet.

Monitoring Functionality
The monitoring functionality counts the number of packets originating from a specific host (H3 in this case). Every time a packet-in event is processed, the script checks if the packet's source MAC address corresponds to H3. If so, it increments a counter.

- **Counting Packets:** The packet counter is incremented within the `packet_in_handler` method whenever a packet from H3 is processed. This counter only increases during packet-in events, which means it counts the number of times the switch had to ask the controller what to do with a packet from H3.

Interaction Between Firewall and Learning Switch

The script combines the logic of a learning switch with the additional firewall and monitoring logic. Here's how they interact:

- **Learning Switch Logic:** The script maintains the functionality of a learning switch by learning the port association of MAC addresses as packets are observed. When the source MAC is new, it's added to the MAC-to-port table. If the destination MAC is already known, a flow is installed in the switch to handle future packets directly, bypassing the controller.

- **Firewall Logic:** Before proceeding with the learning switch logic, the script first checks if the packet should be dropped according to the firewall rules. If a rule is triggered, the packet is dropped, and no learning switch actions are taken for that packet.
- **Monitoring Logic:** Concurrently, the script checks if the packet is from H3 for monitoring purposes. This is independent of whether the packet is dropped or forwarded.

Part 3

The load balancer will direct traffic destined for a virtual IP (VIP), in this case, 10.0.0.42, to actual server IPs in a round-robin fashion.

- **Handle ARP Requests:** The load balancer must respond to ARP requests for the VIP with the MAC address of the load balancer (or switch), so that hosts send the traffic to the load balancer.
- **Round-Robin Distribution:** When the first request comes in for the VIP, the load balancer decides to which server it will forward the request. The next request will be sent to the next server in the list, and so on, cycling through the servers.
- **Flow Rule Installation:** The load balancer installs flow rules in the switch to handle the redirection. These rules will have a hard timeout so that subsequent requests can be distributed to the next server in the round-robin.